Parameterized Complexity of Generalized Domination Problems on Bounded Tree-Width Graphs

Mathieu Chapelle*

LIFO, Université d'Orléans, BP-6759, F-45067 Orléans Cedex 2, France
mathieu.chapelle@univ-orleans.fr

Abstract. The concept of generalized domination unifies well-known variants of domination-like problems. A generalized domination (also called $[\sigma, \rho]$-Dominating Set) problem consists in finding a dominating set for which every vertex of the input graph is satisfied, given two sets of constraints σ and ρ. Very few problems are known to be W[1]-hard when restricted to graphs of bounded tree-width. We exhibit here a large new (infinite) collection of W[1]-hard problems parameterized by the tree-width of the input graph, that is $\exists[\sigma, \rho]$-Dominating Set when σ is a set with arbitrarily large gaps between two consecutive elements and ρ is cofinite (and an additional technical constraint on σ).

1 Introduction

Motivation. Parameterized complexity is a recent theory introduced by Downey and Fellows (see *e.g.* [6, 7] for surveys). This theory undelines the connection between a parameter (different from the usual size of the input) and the complexity of a given problem, and allows to study more precisely its complexity. A problem is said to be FPT (fixed-parameter tractable) parameterized by a parameter k if it can be solved in $\mathcal{O}(f(k) \cdot \text{poly}(n))$ time, for a computable function f and a polynomial p, where n is the size of the input. Parameterized intractable problems are at least W[1]-hard, where W[1] is one of the most important class of parameterized complexity and believed to be strictly including the class FPT (see *e.g.* [6, 7]).

In classical computational complexity, the usual considered parameter is the size of the input graph. From the point of view of parameterized complexity, one can consider several different parameters, *e.g.* the size of the dominating set or the tree-width of the input graph, as the parameter on which the complexity of the problem may depend.

In this article, we study the parameterized complexity of *generalized domination*, also known as $\exists[\sigma, \rho]$-Dominating Set, introduced by Telle [15, 16]. Let σ, ρ be two fixed subsets of $\mathbb{N}$ (throughout this paper $\mathbb{N}$ denotes the set of nonnegative integers while $\mathbb{N}^*$ denotes the set of positive integers). The problem is defined as follows:

$\exists[\sigma, \rho]$-Dominating Set
Input: A graph $G = (V, E)$.
Question: Is there a set $D \subseteq V$ such that for every $v \in D$, $|N(v) \cap D| \in \sigma$, and for every $v \notin D$, $|N(v) \cap D| \in \rho$? If so, D is called a $[\sigma, \rho]$-dominating set.

It is well known that usual optimization problems such as Minimum Dominating Set (minimum $[\sigma, \rho]$-dominating set with $\sigma = \mathbb{N}$ and $\rho = \mathbb{N}^*$) or Maximum Independent Set (maximum $[\sigma, \rho]$-dominating set with $\sigma = \{0\}$ and $\rho = \mathbb{N}$) are **NP**-hard. When dealing with generalized domination, in many cases the problem of finding any $[\sigma, \rho]$-dominating set

* Research partially supported by the French National Research Agency (ANR) project AGAPE.

is already **NP**-hard. Thus one usually considers the existence problem of finding any $[\sigma,\rho]$-dominating set in a given graph, as well as the optimization problems min-$[\sigma,\rho]$-Dominating Set and max-$[\sigma,\rho]$-Dominating Set asking for a dominating set of minimum or maximum size respectively. In this paper, we only consider existence problems unless otherwise stated.

Many difficult (*i.e.* **NP**-hard) well-known problems become efficiently tractable when restricted to graphs of bounded tree-width. This decomposition into a tree-like structure of the input graph allows to write algorithms which efficiently solve the considered **NP**-hard problems. A natural question is whether this tree-like structure can be used to solve every $[\sigma,\rho]$-Dominating Set problems in FPT time parameterized by the tree-width of the input graph.

In this article, we give a more accurated picture of the parameterized complexity of $\exists[\sigma,\rho]$-Dominating Set: we show that for (infinitely) many cases of σ and ρ, $\exists[\sigma,\rho]$-Dominating Set becomes W[1]-hard when parameterized by the tree-width of the input graph.

As one can easily prove (see Appendix C) that $\exists[\sigma,\rho]$-Dominating Set as well as minimization and maximization versions are in XP whenever σ and ρ are recursive sets for which the membership of any integer t can be decided in polynomial time, determining the exact parameterized complexity class of this problem for special cases of σ and ρ is of particular interest. Also note that it can easily be proven (see Appendix C) that whenever σ and ρ are recursive sets (with no restriction), the problem k-$[\sigma,\rho]$-Dominating Set is FPT when parameterized both by the tree-width of the input graph and the maximum size k of a solution.

Related work. The $\exists[\sigma,\rho]$-Dominating Set problem has been extensively studied since its introduction by Telle [15, 16] (see also *e.g.* [10, 13, 7]).

Several results have been found on the classical computational complexity of some cases of $\exists[\sigma,\rho]$-Dominating Set, on general graphs (see *e.g.* [15, 16]), and on some classes of graphs such as bounded tree-width graphs [15, 16, 17] or chordal graphs [8].

From the parameterized complexity point of view, it is well known that most of the usual existence domination problems are W[1]-complete or W[2]-complete on general graphs (see *e.g.* [4, 5]). In an attempt to unify these results, Golovach *et al.* [9] have shown that, when σ and ρ are both finite sets, k-$[\sigma,\rho]$-Dominating Set is W[1]-complete when parameterized by the size k of the expected $[\sigma,\rho]$-dominating set.

For bounded tree-width graphs, van Rooij *et al.* [7] give an $\mathcal{O}^*(s^{tw})$ time algorithm for any $\exists[\sigma,\rho]$-Dominating Set problem with σ and ρ finite or cofinite sets, where s is the minimal number of states needed to recognize σ and ρ, improving a result by Telle *et al.* [17].

One may wonder whether every problem solvable in $\mathcal{O}(n^{\mathrm{poly}(tw)})$ parameterized by tree-width (*i.e.* in XP, see *e.g.* [6, 7]) is also solvable in FPT time for the same parameter. The answer is no, and Lokshtanov *et al.* [11] have shown that k-Equitable Coloring and k-Capacitated Dominating Set are W[1]-hard when parameterized by the size of the expected solution plus the tree-width of the input graph. We will use these results to prove the W[1]-hardness of $\exists[\sigma,\rho]$-Dominating Set parameterized by the tree-width of the input graph for (infinitely) many cases of σ and ρ.

Our result. In this paper, we focus mainly on σ, the constraints of vertices which are in the $[\sigma,\rho]$-dominating set, and study the consequence of considering a non-periodic set σ. We exhibit a large new (infinite) collection of W[1]-hard domination problems parameterized by the tree-width of the input graph, that is $\exists[\sigma,\rho]$-Dominating Set when σ is a set with

arbitrarily large gaps between two consecutive elements (such that a gap of length t is at distance $\mathrm{poly}(t)$ in σ, see Section 3) and ρ is cofinite.

Theorem 1. *Let σ be a set with arbitrarily large gaps between two consecutive elements (such that a gap of length t is at distance* $\mathrm{poly}(t)$ *in σ), and let ρ be cofinite. Then the problem $\exists[\sigma,\rho]$-*Dominating Set *is* W[1]*-hard when parameterized by the tree-width of the input graph.*

Lots of natural well-known infinite sets of integers verify the condition on σ given below, *e.g.* the positive powers of $\alpha \geq 2$ (*e.g.* for $\alpha = 3$, the set $\{3, 9, 27, 81, \ldots\}$), the set of all prime numbers ($\{2, 3, 5, 7, 11, 13, \ldots\}$), or the Fibonacci numbers ($\{1, 2, 3, 5, 8, 13, \ldots\}$). On the other hand, this result doesn't work for infinite sets with bounded gaps, *e.g.* the set of all integers excepted the multiple of $\alpha \geq 2$ (for *e.g.* $\alpha = 3$, the set $\{0, 1, 2, 4, 5, 7, 8, 10, \ldots\}$), or the set of integers with gaps corresponding to the digits of a real β with infinite digits (for *e.g.* $\beta = 3.1415\ldots$, the set $\{3, 5, 10, 12, 18, \ldots\}$).

By a result from Courcelle *et al.* [3, 4], one can prove that if σ and ρ are both ultimately periodic sets (see *e.g.* [6]), then any problem $[\sigma,\rho]$-Dominating Set (existence, minimization and maximization) is FPT when parameterized by the tree-width of the input graph (the proof of this result is postponed to Appendix B).

Associated with our W[1]-hardness result, we are getting closer to a complete dichotomy of the parameterized complexity of $\exists[\sigma,\rho]$-Dominating Set parameterized by tree-width.

2 Preliminaries

Graphs. Let $G = (V, E)$ be a finite undirected n-vertex m-edge graph without loops nor multiple edges. $V(G)$ (or simply V if its clear from the context) denotes the *set of vertices* of the graph G, while $E(G)$ (or simply E) denotes the *set of edges*. For two vertices $x, y \in V$, we denote an edge betwen x and y by xy. For a vertex $v \in V$, $N(v) = \{u \mid uv \in E\}$ denotes the open neighborhood of v, while $N[v] = N(v) \cup \{v\}$ denotes its closed neighborhood. For a subset $S \subseteq V$, $N[S] = \bigcup_{v \in S} N[v]$ denotes the closed neighborhood of S.

A vertex v is *dominated* by a vertex u if $v \in N[u]$, and it is *dominated* by a set $S \subseteq V$ if $v \in N[S]$. A subset of vertices $S \subseteq V$ is called a *dominating set*[1] if every vertex of G is dominated by S. A vertex v which is added to the dominating set is said to be *selected*, while a vertex which is not added to the dominating set is said to be *non-selected*.

The *incidence graph* $I(G)$ of a graph G is a bipartite graph with $V(G) \cup E(G)$ as set of vertices, and for two vertices e, v of $I(G)$, with $e \in E(G)$ and $v \in V(G)$, e is adjacent to v in $I(G)$ if e is incident to v in G.

Tree-width. A *tree-decomposition* [1] of a graph G is a rooted tree T in which each node $i \in T$ has an assigned set of vertices $X_i \subseteq V(G)$ (called *bag*), such that (1) every vertex $v \in V(G)$ appears in at least one *bag* X_i of T, (2) every edge $uv \in V(G)$ has its both ends appearing in a same *bag* X_j of T, and (3) for every vertex $v \in V(G)$, the *bags* containing v induce a connected subtree of T. The width of a tree-decomposition is the size of the largest bag of T minus one, *i.e.* $\max_{i \in T} |X_i| - 1$. The *tree-width* of a graph G is then the minimum width over all tree-decompositions of G.

[1] We suppose that $0 \notin \rho$, as otherwise $S = \emptyset$ would be a trivial solution.

3 Proof of Theorem 1

To prove Theorem 1, we will reduce from k-Capacitated Dominating Set which is known to be W[1]-complete when parameterized by the tree-width of the input graph plus the size of the expected capacitated dominating set [11]:

k-Capacitated Dominating Set
Input: A graph $G = (V, E)$ of tree-width tw, a function $\mathrm{cap} : V \to \mathbb{N}$, and a positive integer k.
Parameter: $k + tw$.
Question: Does G admit a set S of cardinality at most k and a *domination function* dom associating to each vertex $v \in S$ a set $\mathrm{dom}(v) \subseteq V \setminus S$ of at most $\mathrm{cap}(v)$ vertices, such that every vertex $w \in V \setminus S$ is in $\mathrm{dom}(v)$ for some $v \in S$? If so, (S, dom) is called a k-capacitated dominating set.

Let σ be a set with arbitrarily large gaps between two consecutive elements (such that a gap of length t is at distance $\mathrm{poly}(t)$ in σ), and let ρ be cofinite. We define $q_0 = \min_{q\in\rho}\{q \mid \forall r \geq q, r \in \rho\}$. We give in the following an *fpt*-reduction from k-Capacitated Dominating Set to $\exists[\sigma, \rho]$-Dominating Set (the size of the constructed graph is polynomial in the size of the input graph, and the tree-width is "almost" preserved), thus proving this latter problem is W[1]-hard when parameterized by the tree-width of the input graph.

For readability reasons, we suppose that $\min \rho \geq 2$ (hence $q_0 \geq 2$) and $\min \sigma \geq 1$. The construction, gadgets and proofs can easily be adapted to the extremal cases when $\min \rho \geq 1$ or $\min \sigma = 0$.[2]

Some functions on $\boldsymbol{\sigma}$. For the construction of our *fpt*-reduction, we suppose that we are given some computable functions on σ:

- $\Gamma_-(x, q)$: returns the lowest element p of σ, greater than q, for which there are at least x integers *before* p which are not elements of σ;
- $\Gamma_+(x, q)$: returns the lowest element p of σ, greater than q, for which there are at least x integers *after* p which are not elements of σ;
- $\Gamma_0(q)$: returns the lowest element p of σ greater than q.

Those functions will allow us to find some gaps of particular length in σ used in the construction of our gadgets for the *fpt*-reduction.

However, we need a technical condition on such gaps. The construction of some of our gadgets (gadgets $\mathcal{C}$ and $\mathcal{L}$, see below) will require gaps of length depending on the number of vertices in the input graph. For this purpose, we suppose that a gap of length t can be found at distance $\mathrm{poly}(t)$ in σ, that is $\Gamma_-(t, q)$ (for some q) is polynomial in t.

***fpt*-reduction from k-Capacitated Dominating Set.** Let $I(G)$ be the incidence graph of G. We note $N_I(v)$ the neighborhood of v in $I(G)$, corresponding to the set of edges incident to v in G. An *original-vertex* in $I(G)$ corresponds to a vertex in G, while an *edge-vertex* in $I(G)$ corresponds to an edge in G. For an $[\sigma, \rho]$-dominating set D, a vertex v is said to be *selected* if $v \in D$.

Given an instance (G, cap, k) of k-Capacitated Dominating Set, where G is of tree-width at most tw and cap is the function of capacities on the vertices of G, we construct an

[2] Recall that we suppose $0 \notin \rho$, as otherwise $S = \emptyset$ would be a trivial $[\sigma, \rho]$-dominating set.

instance H of $\exists[\sigma,\rho]$-Dominating Set with H of bounded tree-width, such that G admits a k-Capacitated Dominating Set if and only if H admits a $[\sigma,\rho]$-dominating set. The idea for the *fpt*-reduction is that using the functions Γ, we can find a *gap* in σ of size greater than k (the parameter of k-Capacitated Dominating Set), and use this *gap* to control the number of selected neighbors with respect to the capacity of each selected vertex.

To construct H, we start with a copy of $I(G)$, and we add some gadgets on original-vertices and edge-vertices of $I(G)$ (see Figure 1). We add gadgets *capacity*, *satisfiability* and *domination* on each original-vertex of $I(G)$, with gadget *domination* linked to each edge-vertex neighboring the corresponding original-vertex, and a gadget *edge-selection* on each edge-vertex of $I(G)$. We also add a global gadget *limitation* with one *central* vertex linked to each original-vertex of $I(G)$ (see Figure 2).

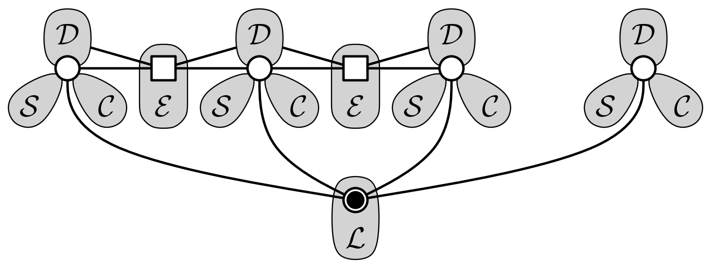

Fig. 1. The overall construction of the graph H. White vertices are vertices of $I(G)$: white circles are original-vertices, and white squares are edge-vertices.

Suppose G admits a k-capacitated dominating set. We explain how this k-capacitated dominating set S of G will result into a $[\sigma,\rho]$-dominating set D of H.

If a vertex of G is in S, then the corresponding original-vertex of H will be in D. For each vertex $u \in V(G) \setminus S$ which is dominated by a vertex $v \in V(G)$, the edge-vertex e in $I(G)$ representing the edge between u and v in G will also be in D. The gadget *limitation* will ensure that no more than k original-vertices of H are selected into D, and hence will translate into a k-capacitated dominating set of G.

The other vertices of H (those which are not in $I(G)$) are vertices of the several gadgets.

Into the different gadgets, there will be two kind of vertices. If a solution exists, *forced* vertices will be forced to be selected, while *choosable* vertices will always be satisfied no matter if they are selected or not.

We now describe the gadgets used for the *fpt*-reduction (see Figure 2):

- *force gadget* ($\mathcal{F}$), one for each vertex to be *forced*. This technical gadget forces a given vertex of H to be selected if a solution exists.

  For a vertex $w \in V(H)$ we want to force, we add a clique with $\min\sigma$ vertices linked to w. Let $\alpha_{\sigma,\rho} = \min_{p\in\mathbb{N}}\{p \in \sigma \ \wedge \ p \in \rho \ \wedge \ p + \min\sigma + 1 \in \sigma\}$ and $\beta_{\sigma,\rho} = \min_{p'\in\mathbb{N}}\{p - 1 \in \rho \ \wedge \ p + 1 \in \sigma\}$. As ρ is cofinite, $\sigma \cap \rho \neq \emptyset$ and hence $\alpha_{\sigma,\rho}$ always exists. We also add a clique with $\alpha_{\sigma,\rho}$ vertices linked to w, and a clique with $\beta_{\sigma,\rho}$ vertices linked to every vertex of the former clique.

- *domination gadget* ($\mathcal{D}$), one for each original-vertex in $I(G)$. This gadget ensures that an original-vertex of $I(G)$ is either selected, or has at least one selected neighbor in $I(G)$, *i.e.* the selected vertices form a dominating set in $I(G)$.
  For each original-vertex $v \in I(G)$, we add a (non-selectable) vertex v' linked to v and to each edge-vertex $e \in N_I(v)$, $q_0 - 2$ independent *forced* vertices linked to v', and an extra *forced* vertex linked to v' with $\Gamma_+(1,0)$ independent *forced* neighbors and to a clique with $\min\sigma$ vertices.
- *edge-selection gadget* ($\mathcal{E}$), one for each edge-vertex in $I(G)$. The selected edge-vertices will correspond to the *domination function* of the k-Capacitated Dominating Set problem we reduce from. This gadget ensures that each selected edge-vertex in $I(G)$ has at least one selected neighbor (vertex of G) in $I(G)$).
  For each edge-vertex $e \in I(G)$, we add $\Gamma_-(1, q_0+1) - 1$ independent *forced* vertices linked to e.
- *capacity gadget* ($\mathcal{C}$), one for each original-vertex in $I(G)$. This gadget ensures that a selected original-vertex v of $I(G)$ has at most $\mathrm{cap}(v)$ selected neighbors in $I(G)$. Moreover, this gadget allows a selected original-vertex to have a valid number of selected neighbors in H with respect to σ.
  For each original-vertex $v \in I(G)$, we add $\Gamma_+(\deg_G(v)+q_0, \mathrm{cap}(v)) - \mathrm{cap}(v) - 1$ independent *forced* vertices linked to v, and $\mathrm{cap}(v)$ independent *choosable* vertices linked to v with $\Gamma_0(q_0+1) - 1$ independent *forced* neighbors each.
- *satisfiability gadget* ($\mathcal{S}$), one for each original-vertex in $I(G)$. This gadget allows any non-selected vertex of $V(G)$ to have a valid number of selected neighbors in H with respect to ρ.
  For each original-vertex $v \in I(G)$, we add q_0 independent *choosable* vertices with $\Gamma_0(q_0)$ independent *forced* neighbors each.
- *limitation gadget* ($\mathcal{L}$), one for the whole graph H. This gadget limits the number of selected original-vertices in $I(G)$ to at most k vertices, where k is the parameter of the original k-Capacitated Dominating Set we reduce from.
  We add one *central forced* vertex c linked to every original-vertex of $I(G)$ and to a clique with $\min\sigma$ vertices, $\Gamma_+(|V(G)|, k) - k$ independent *forced* vertices linked to c, and k independent *choosable* vertices linked to c with $\Gamma_0(q_0) - 1$ independent *forced* neighbors each.

Proof of correctness.

Lemma 1. *All vertices of all gadgets,* i.e. forced *and* choosable *vertices, are always satisfied, and the gadgets play their respective roles as defined above. Moreover, if H admits a $[\sigma, \rho]$-dominating set, then all original-vertex and every edge-vertex of $I(G)$ are satisfied.*

Proof. Due to space restrictions, the proof is postponed to Appendix A.

Lemma 2. *The input graph G admits a k-capacitated dominating set if and only if the constructed graph H admits a $[\sigma, \rho]$-dominating set.*

Proof. For the "if" part, let D be a $[\sigma, \rho]$-dominating set of H. Let H_0 be the vertices of H corresponding to vertices of G, and let $D_1 \subseteq D$ (resp. $D_2 \subseteq D$) be the selected vertices in H which arise from vertices of G (resp. from edges of G). Note that $D_1 \subseteq H_0$.

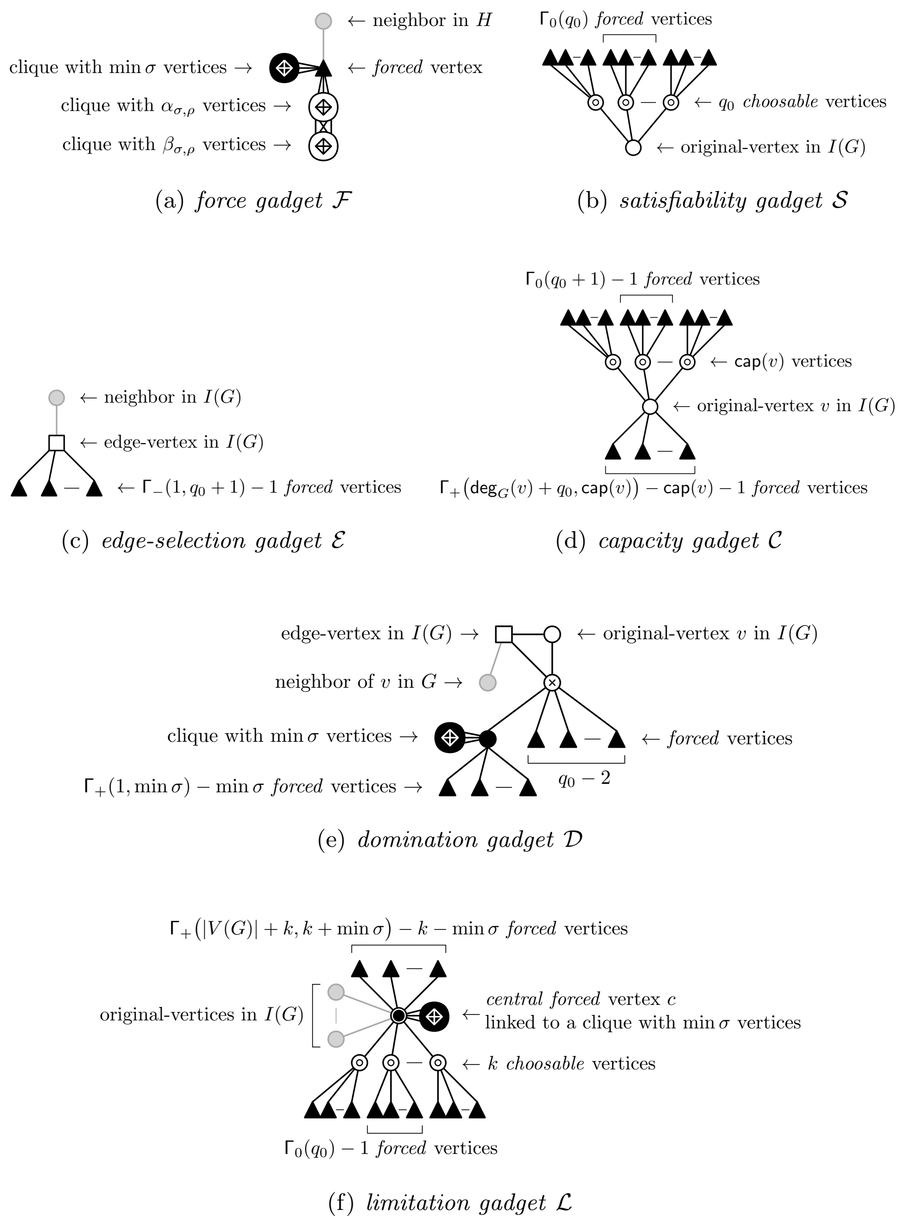

Fig. 2. The gadgets used for the *fpt*-reduction from k-Capacitated Dominating Set to $\exists[\sigma,\rho]$-Dominating Set. Black triangular vertices and black-disk vertex are *forced*, circled vertices are *choosables*, and crossed vertex is *non-selectable*.

By gadget *domination*, each vertex in H_0 is either selected (*i.e.* is in D_1) or has a selected neighbor in D_2, and hence $D_1 \cup D_2$ forms a dominating set in $I(G)$. By gadget *edge-selection*, each vertex in D_2 has at least one selected neighbor in D_1. By gadget *capacity*, each selected vertex v in H_0 (*i.e.* $v \in D_1$) has at most $\mathrm{cap}(v)$ selected neighbors in D_2, and $|N_H(v) \cap S| \in \sigma$. For a selected vertex v, we set $\mathrm{dom}(v) = \{u \mid u \in N_H(v) \cap D_2\}$. By gadget *satisfiability*, each non-selected vertex in H_0 (*i.e.* $v \in H_0 \setminus D_1$) has at least q_0 selected neighbors in H, and hence $|N_H(v) \cap S| \in \rho$. Finally, by gadget *limitation*, $|D_1| \leq k$.

Then (D_1, dom) is a k-capacitated dominating set of G, where dom is the *function of domination* of vertices in G.

For the "only if" part, let (S, dom) be a k-capacitated dominating set of G, where dom is the *domination function* of vertices in G. We construct a $[\sigma, \rho]$-dominating set D of H. Let $I(G)$ be the incidence graph of G.

For each selected vertex $v \in S$, and each dominated vertex $u \in V(G)$ such that $u \in \mathrm{dom}(v)$, add the edge-vertex $e \in V(I(G))$ to D, where e corresponds to the edge uv in G. Then the vertices of $I(G)$ as well as the vertices of gadgets *domination* and *edge-selection* are satisfied, and as every other gadget contains only *forced* and *choosable* vertices which are always satisfied, H admits a $[\sigma, \rho]$-dominating set containing D. □

Lemma 3. *The reduction from* k-Capacitated Dominating Set *to* $\exists[\sigma, \rho]$-Dominating Set *is* FPT. *More precisely, the graph H can be constructed in* $\mathrm{poly}(|V(G)|)$ *time, and* $tw(H) \leq 4tw(G) + \mathrm{const}(\sigma, \rho)$.

Proof. Let σ and ρ be fixed. Let G be the input graph of k-Capacitated Dominating Set and H be the constructed graph for $\exists[\sigma, \rho]$-Dominating Set.

First, we prove that the size of H is polynomial in the size of G. The *limitation* gadget is created once for the whole graph, and the *capacity* gadget is created once for each original-vertex. The cardinalities of those two gadgets depend on the Γ functions, and a polynomial in the size of G due to our technical constraint on σ (see Section 3). The technical *force* gadget contains $\min \sigma + 1 + \alpha_{\sigma,\rho} + \beta_{\sigma,\rho}$ vertices for each *forced* vertex. The other gadgets are created once for each original-vertex or edge-vertex of $I(G)$, and each has a cardinality depending on the Γ functions and on σ and ρ. Hence the overall number of vertices in H depends only on σ and ρ which are fixed and hence constants, and polynomialy on the number of vertices and edges of G due to our technical constraint on σ. The reduction is then polynomial in the size of the input graph plus the parameter k.

We now prove that the tree-width of H is polynomial in the tree-width of G. Let $T(I(G))$ be an optimal tree-decomposition of $I(G)$, whose tree-width is at most the tree-width of G. We explain how one can construct a tree-decomposition of H of width bounded by the tree-width of $I(G)$.

If we consider each *force* gadget as one vertex, then every gadget is a tree. Moreover each gadget has only one vertex (the root of the gadget) which is linked to some original-vertices and/or edge-vertices of $I(G)$: the root of *domination* gadget is linked to one original-vertex of $I(G)$ and every edge-vertex adjacent to this original-vertex, the roots of *satisfiability* and *capacity* are linked to one original-vertex, the root of *edge-selection* gadget is linked to one edge-vertex, and the root of *limitation* gadget is linked to every original-vertex of $I(G)$ (see Figure 1). Starting from the tree-decomposition $T(I(G))$ of $I(G)$, we link the tree-decomposition of each gadget to a bag containing one of the original-vertices or edge-vertices of $I(G)$ to which the gadget is linked. We then add the root of each gadget to all the bags of

the tree-decomposition containing its neighbors in $I(G)$. Thus for a given vertex of $I(G)$ we add at most 3 vertices to the bags containing this vertex.

Now consider the *force* gadget. It contains a clique with $\min\sigma + 1$ vertices and a clique with $\alpha_{\sigma,\rho} + \beta_{\sigma,\rho}$ vertices, and hence is of tree-width at most $\alpha_{\sigma,\rho} + \beta_{\sigma,\rho}$ (the size of the biggest clique). Hence all the other gadgets, which are only trees excepted for their *forced* vertices, are also of tree-width at most $\alpha_{\sigma,\rho} + \beta_{\sigma,\rho}$ (the tree-width of the *force* gadget). As for a given vertex of $I(G)$ at most 3 vertices are added to the bags containing this vertex, the overall tree-decomposition of H is of width $tw(H) \leq 4tw(G) + \alpha_{\sigma,\rho} + \beta_{\sigma,\rho}$, where $\alpha_{\sigma,\rho} + \beta_{\sigma,\rho}$ is a constant as σ and ρ are fixed. Hence the reduction is FPT with respect to the parameter, that is the tree-width of the input graph G. □

4 Conclusion

We have proven that the $\exists[\sigma,\rho]$-Dominating Set problem parameterized by the tree-width of the input graph becomes W[1]-hard when σ can have arbitrarily large gaps between two consecutive elements (such that a gap of length t is at distance poly(t) in σ) and ρ is cofinite.

This result gives a large new (infinite) collection of domination problems being W[1]-hard when parameterized by the tree-width of the input graph. Associated with a result from Courcelle *et al.* [3, 4], which allows one to prove (see Appendix B) that when σ and ρ are ultimately periodic sets the problem $\exists[\sigma,\rho]$-Dominating Set as well as minimization and maximization versions are FPT when parameterized by the tree-width of the input graph, we are getting closer to a complete dichotomy of the parameterized complexity of $\exists[\sigma,\rho]$-Dominating Set parameterized by tree-width.

This naturally requests further investigations onto the parameterized complexity of this problem for other cases of sets σ and ρ. In particular, what is the parameterized complexity of $\exists[\sigma,\rho]$-Dominating Set when σ is recursive with bounded gaps between two consecutive elements but not ultimately periodic? Can we circumvent the technical constraint on σ which requires a gap of length t to be at distance poly(t) in σ?

Acknowledgement. The author wishes to thank Dieter Kratsch, Mathieu Liedloff and Ioan Todinca for constructive remarks on some parts of this work.

References

1. Bodlaender, H.L.: A partial k-arboretum of graphs with bounded treewidth. Theoretical Computer Science **209**(1) (1998) 1–45
2. Courcelle, B.: The expression of graph properties and graph transformations in monadic second-order logic. In: Handbook of Graph Grammars and Computing by Graph Transformation. Volume 1., River Edge, NJ, World Scientific (1997) 313–400
3. Courcelle, B., Makowsky, J.A., Rotics, U.: On the fixed parameter complexity of graph enumeration problems definable in monadic second-order logic. Discrete Applied Mathematics **108**(1–2) (2001) 23–52
4. Downey, R.G., Fellows, M.R.: Fixed-parameter tractability and completeness. I. basic results. SIAM Journal on Computing **24**(4) (1995) 873–921
5. Downey, R.G., Fellows, M.R.: Fixed-parameter tractability and completeness. II. on completeness for W[1]. Theoretical Computer Science **141** (1995) 109–131
6. Downey, R.G., Fellows, M.R.: Parameterized complexity. Springer (1999)
7. Flum, J., Grohe, M.: Parameterized complexity theory. Springer (2006)

8. Golovach, P., Kratochvíl, J.: Computational complexity of generalized domination: a complete dichotomy for chordal graphs. In: WG'07, Proceedings of the 33rd International Workshop on Graph-Theoretic Concepts in Computer Science. Number 4769 in Lecture Notes in Computer Science, Dornburg, Germany, Springer (June 2007) 1–11
9. Golovach, P.A., Kratochvíl, J., Suchý, O.: Parameterized complexity of generalized domination problems. In: WG'09, Proceedings of the 35th International Workshop on Graph-Theoretic Concepts in Computer Science. Volume 5911 of Lecture Notes in Computer Science. (2010) 133–142
10. Golovach, P.A., Villanger, Y.: Parameterized complexity for domination problems on degenerate graphs. In: WG'08, Proceedings of the 34th International Workshop on Graph-Theoretic Concepts in Computer Science. Volume 5344 of Lecture Notes in Computer Science., Springer (2008) 195–205
11. Lokshtanov, D.: New methods in parametrized algorithms and complexity. PhD thesis, University of Bergen (April 2009)
12. Matos, A.B.: Periodic sets of integers. Theoretical Computer Science **127** (1994) 287–312
13. Raman, V., Saurabh, S., Srihari, S.: Parameterized algorithms for generalized domination. In: COCOA'08, Proceedings of the 2nd International Conference on Combinatorial Optimization and Applications. Volume 5165 of Lecture Notes in Computer Science., Springer (2008) 116–126
14. van Rooij, J.M., Bodlaender, H.L., Rossmanith, P.: Dynamic programming on tree decompositions using generalised fast subset convolution. In: ESA'09, Proceedings of the 17th Annual European Symposium on Algorithms. Volume 5757 of Lecture Notes in Computer Science., Copenhagen, Denmark, Springer (September 2009) 566–577
15. Telle, J.A.: Complexity of domination-type problems in graphs. Nordic Journal of Computing **1** (1994) 157–171
16. Telle, J.A.: Vertex partitioning problems: characterization, complexity and algorithms on partial k-trees. PhD thesis, University of Oregon (1994)
17. Telle, J.A., Proskurowski, A.: Algorithms for vertex partitioning problems on partial k-trees. SIAM Journal on Discrete Mathematics **10**(4) (1997) 529–550

A Proof of Lemma 1 (correctness of the gadgets)

We prove that inside each gadget, every vertex (*forced* or *choosable*) is always satisfied, and that the gadgets play their respective roles.

Inside the gadgets, there are two kind of vertices: vertices that are *forced* (*i.e.* must be selected), and vertices that are *choosable* (*i.e.* can be selected). Each *forced* vertex is forced to be selected by the *force* gadget (see below), while each *choosable* vertex has some forced neighbors which allows it to be satisfied no matter if it is selected or not.

For gadget *force*, let v be the vertex to be *forced*. It has a clique with $\min \sigma$ vertices and a clique with α vertices as neighbors in the gadget. As we look for a $[\sigma, \rho]$-dominating set with $0 \notin \rho$, at least one of the vertices of the clique with $\min \sigma$ vertices or v must be selected. If v is selected, each vertex of the clique has one selected neighbor, and as $1 \notin \rho$ (recall that for readability reasons we supposed that $\min \sigma \geq 1$ and $\min \rho \geq 2$), at least one vertex of the clique must be selected. But if a vertex of the clique is selected, then it needs at least $\min \sigma$ selected neighbors, and hence every vertex of the clique and v must be selected. Thus every vertex of the clique and v must be selected, which implies that v must be selected. Moreover, recall that $\alpha_{\sigma,\rho} = \min_{p \in \mathbb{N}} \{p \in \sigma \wedge p \in \rho \wedge p{+}\min \sigma{+}1 \in \sigma\}$ and $\beta_{\sigma,\rho} = \min_{p' \in \mathbb{N}} \{p{-}1 \in \rho \wedge p{+}1 \in \sigma\}$, and there is a clique with $\beta_{\sigma,\rho}$ vertices linked to every vertex of the clique with $\alpha_{\sigma,\rho}$ vertices. If the neighbor of v in H is not selected, then every vertex of the clique with $\beta_{\sigma,\rho}$ vertices can be selected, and every vertex of the gadget will be satisfied. If the neighbor of v in H is selected, then every vertex of the clique with $\alpha_{\sigma,\rho}$ vertices can be selected, and every vertex of the gadget will be satisfied. Indeed, $\alpha_{\sigma,\rho}$ is defined in order to satisfy every vertex of the gadget if the neighbor of v in H is selected, while $\beta_{\sigma,\rho}$ is defined in order to satisfy every vertex of the gadget if the neighbor of v in H is not selected.

For gadget *satisfiability*, let v be the controlled non-selected vertex. It has q_0 independent *choosable* neighbors in the gagdet which can be selected as they have $\Gamma_0(q_0)$ *forced* neighbors each. Thus v must always have at least q_0 selected vertices in H, and hence it will have a valid number of selected neighbors in H.

For gadget *edge-selection*, let e be the controlled edge-vertex. It has $\Gamma_-(1, q_0+1)-1$ *forced* neighbors in the gadget. If e is not selected, then it has $\Gamma_-(1, q_0+1)-1 \in \rho$ *forced* neighbors in the gadget, and hence has a valid number of selected neighbors in H. If e is selected, then it needs at least one more selected neighbor in H (which is a vertex of G) as $\Gamma_-(1, q_0+1)-1 \notin \sigma$ while $\Gamma_-(1, q_0+1) \in \sigma$.

For gadget *capacity*, let v be the selected vertex whose capacity is controlled. It has $\Gamma_+(\deg_G(v)+q_0, \mathrm{cap}(v)) - \mathrm{cap}(v)$ *forced* neighbors in the gadget (including the *central forced* vertex of gadget *limitation*), and hence can have at most $\mathrm{cap}(v)$ more selected vertices in H. The only other neighbors it has in H are edge-vertices from $I(G)$, and $\mathrm{cap}(v)+q_0$ independent *choosable* neighbors in the gadgets *capacity* and *satisfiability* that can be selected if v has less that $\mathrm{cap}(v)$ selected neighbors in $I(G)$. The independent *choosable* neighbors in gadget *capacity* have always a valid number of selected neighbors in H. Indeed, for any of those *choosable* neighbors, one can be selected if v is selected and hence it has $\Gamma_0(q_0+1) \in \sigma$ selected neighbors, and if one is not selected, then it has at least $\Gamma_0(q_0+1) \in \rho$ selected neighbors. Finally, v can have exactly $\Gamma_+(\deg_G(v)+q_0, \mathrm{cap}(v)) \in \sigma \cap \rho$ selected neighbors in H, and hence it will have a valid number of selected neighbors in H.

For gadget *domination*, let v be the vertex of G whose domination state is controlled. The corresponding vertex v' in the gadget has q_0-2 *forced* neighbors in the gadget, and one extra *forced* neighbor (forced by the clique with $\min \sigma$ vertices as in the *force* gadget) with $\Gamma_+(1,0)$

forced neighbors which forbid v' from being selected (as otherwise this extra vertex would have $\Gamma_+(1,0)+1 \notin \sigma$ selected neighbors). Thus v' has exactly $q_0 - 1 \notin \rho$ selected neighbors and can't be selected, so it needs at least one more selected neighbor in H (which will be in $I(G)$).

For gadget *limitation*, let c be the *central forced* vertex which limits the number of selected vertices in G to at most k. It is forced by the clique with $\min \sigma$ vertices as in the *force* gadget. It has $\Gamma_+(|V(G)|+k, k+\min\sigma) - k$ *forced* neighbors in the gadget (including the $\min\sigma$ vertices of the clique to which it is linked), and k independent *choosable* neighbors in the gadget which can be selected if c has less than k selected neighbors in G. Thus c can have at most $\Gamma_+(|V(G)|+k, k+\min\sigma)$ selected neighbors in H, at most k of them being vertices of G as the next element in σ which is greater than $\Gamma_+(|V(G)|+k, k+\min\sigma)$ is greater than the number of vertices in G. Hence c will have a valid number of selected neighbors in H.

B FPT cases

By a result from Courcelle *et al.* [4], problems on graphs of bounded tree-width which are expressible in CMSOL (an extension of MSOL introduced by Courcelle [3]) are solvable in FPT time. One can prove that $\exists[\sigma,\rho]$-Dominating Set is expressible in a classical monadic second-order logic (MSOL) formula for graphs of bounded tree-width when σ and ρ are finite or cofinite sets. Using CMSOL, one can also find a counting monadic second-order logic (CMSOL) formula which expresses $\exists[\sigma,\rho]$-Dominating Set when σ and ρ are ultimately periodic sets. Unfortunately, the generic FPT algorithm given by Courcelle *et al.* [4] is not efficient in practice.

We suppose here that σ and ρ are two ultimately periodic sets. We will give an algorithm which solves $\exists[\sigma,\rho]$-Dominating Set efficiently in FPT time on graphs of bounded tree-width. This algorithm can easily be adapted to also return a dominating set of minimum or maximum size.

Our algorithm will use a bottom-up dynamic programming approach on the nice tree-decomposition of the input graph. As the input graph is of bounded tree-width, we can suppose that such nice tree-decomposition is given (it can be constructed in FPT time [2, 5]). Due to space restriction, we only describe here the operation and time complexity on *join* node, which is the most costly, and give hints for the other nodes.

Some definitions. σ and ρ are two ultimately periodic sets. Thus there exist two unary-language finite deterministic automata of minimum size (in the number of states) which can enumerate the elements of σ and ρ respectively (see [6]). Let p (resp. q) be the size (*i.e.* number of states) of the minimal automaton iteratively enumerating the elements of the set σ (resp. ρ). For convenience, we will denote by $\mathcal{Q}_\sigma = \{\sigma_0, \sigma_1, \ldots, \sigma_p\}$ and $\mathcal{Q}_\rho = \{\rho_0, \rho_1, \ldots, \rho_q\}$ the states of the automata of σ and ρ respectively.

Let $G=(V,E)$ be the input graph, and (T,χ) a nice tree-decomposition of G.

Definition 1 (partial solution). *Let T_i be the subtree of T rooted at the node i of T, and G_i the subgraph of G containing only vertices of G appearing in the bags of T_i. A subset $S_i \subseteq V$ associated to a characteristic of i is called a* partial solution *if every vertex in $V(G_i)\setminus\{v \mid v \in X_i\}$ has a valid number of selected neighbors with respect to σ and ρ.*

Characteristics will contain the current state of each vertex in the corresponding automaton: selected vertices will have states from the automaton of σ, while non-selected vertices

will have states from the automaton of ρ. At the end of the algorithm, the solution of the problem (if one exists) will be found in one of the characteristics of the root of the nice tree-decomposition.

Definition 2 (state of a vertex). *Let $v \in V$ be a vertex of G, i a node of T, S_i a partial solution, and Q_σ (resp. Q_ρ) the set of states of the automaton $\mathcal{A}_\sigma$ (resp. $\mathcal{A}_\rho$) enumerating the elements of σ (resp. ρ). We define the* state of v in X_i, *noted* $\mathrm{s}_i(v)$, *by*

$$\mathrm{s}_i(v) = \begin{cases} \sigma_j \in Q_\sigma & \text{if } v \in S_i,\ 1 \leq j \leq p \\ \rho_k \in Q_\rho & \text{if } v \notin S_i,\ 1 \leq k \leq q \end{cases}$$

To each node of the nice tree-decomposition, the algorithm will associate a collection of *characteristics* which will correspond to different states of the vertices in the given node.

Definition 3 (characteristic of a node). *Let i be a node of T, S_i a partial solution, and $n_i = |X_i| \leq tw(G)+1$. A* characteristic *of the node i is a n_i-plet of states* $(\mathrm{s}_i(v_1), \ldots, \mathrm{s}_i(v_{n_i}))$, *where the state $\mathrm{s}_i(v_j)$ of v_j corresponds to the number of selected neighbors it has in S_i. $c(i)$ denotes the collection of characteristics of the node i.*

For the operation on *join* nodes, we need to define *compatible characteristics.*

Definition 4 (compatible characteristics). *Let i be a* join *node of T, j and k the two child nodes of i, and $\mathcal{A}_\sigma$ and $\mathcal{A}_\rho$ the two automata enumerating the elements of σ and ρ respectively. Remark that $n_i = |X_i| = |X_j| = |X_k|$. Two characteristics $(e_1, \ldots, e_{n_i}) \in c(j)$ and $(f_1, \ldots, f_{n_i}) \in c(k)$ are* compatible *if for every $1 \leq l \leq n_i$, e_l and f_l are states from the same automaton,* i.e. *$e_l \in Q_\sigma \Leftrightarrow f_l \in Q_\sigma$ and $e_l \in Q_\rho \Leftrightarrow f_l \in Q_\rho$.*

Operation on *leaf*, *introduce*, *forget*, and root nodes. The algorithm propagates *partial solutions* from the leaves to the root of the nice tree-decomposition, using *characteristics* on its nodes.

On a *leaf node* introducing the node v, the algorithm creates two characteristics of one element each (for v): one for v selected (its state will be in $\mathcal{A}_\sigma$), and one for v non-selected (its state will be in $\mathcal{A}_\rho$).

On an *introduce node* introducing the node v, the algorithm will duplicate every characteristics of its child, with one for v selected, and the other for v non-selected. In the first case, the state of each vertex in the characteristic is updated, as it then has one more selected neighbor.

On a *forget node* forgetting the node v, the algorithm first keeps only characteristics of its child in which the state of v is valid, *i.e.* is a final state of the corresponding automaton. It then copies each remaining characteristic without the state of v.

On the root node of the nice tree-decomposition, the algorithm searches for a characteristic in which every vertex has a valid state, *i.e.* is a final state of the corresponding automaton.

Operation *join* node. The operation on a *join* node basically consists on merging two *compatible* characteristics from the characteristics of the two child nodes.

Let $i \in \chi$ be a *join* node, and let j and k be its two child nodes. As we use a nice tree-decomposition, $X_i = X_j = X_k$ and $n_i = |X_i| = |X_j| = |X_k|$. The collection $c(i)$ of characteristics of the node i is constructed from the characteristics of j and k by combining two compatible characteristics. Note that a vertex $v \in X_i$ can be dominated by another vertex

$u \in X_i$ in the partial solution, so the merging operation must prevent u from being counted twice (in $\mathrm{s}_j(v)$ and $\mathrm{s}_k(v)$).

A characteristic in $c(i)$ is constructed from each couple of compatible characteristics $(\mathrm{s}_j(v_1), \ldots, \mathrm{s}_j(v_{n_i})) \in c(j)$ and $(\mathrm{s}_k(v_1), \ldots, \mathrm{s}_k(v_{n_i})) \in c(k)$. Because they are compatible, for every $v \in X_i$, the state of v in X_j is in the same automaton as its state in X_k, and hence also will be the state of v state in X_i. Let $N_i = \{u \mid u \in X_i, \mathrm{s}_i(u) \in Q_\sigma\}$ be the set of selected neighbors of v which are counted twice (both in j and in k). Let $\mathrm{s}_k(v) = \sigma_l$. We define $||\mathrm{s}_k(v)|| = l$ if $l \geq |N_i|$, or $||\mathrm{s}_k(v)|| = l + |N_i|$ if $l < |N_i|$. The state of v in X_i is constructed in this way:

$$\mathrm{s}_i(v) = \Delta\Big(\mathrm{s}_j(v), ||\mathrm{s}_k(v)|| - |N_i|\Big), \quad \forall v \in X_i,$$

where $\Delta(\mathrm{s}(v), b)$ is the new state of v, starting at state $\mathrm{s}(v)$ and iterating b time the transition function of the corresponding automaton. Remark that v has at least $|N_j|$ selected neighbors in X_i, so the number of selected neighbors counted by $\mathrm{s}_k(v)$ is at least $|N_i|$.

Proof of correctness and time complexity.

Lemma 4. *The operation on* join *nodes is correct.*

Proof. Let i be a *join* node, and j and k its two children. It suffices to prove that given two compatible characteristics from j and k with partial solutions S_j and S_k, the partial solution S_i associated to the constructed characteristic in i is valid, *i.e.* the vertices in $V(G_i) \setminus \{v \mid v \in X_i\}$ have a valid number of selected neighbors counted by their state in the corresponding automaton.

Indeed, $S_i = S_j \cup S_k$. Moreover, the operation on i does not modify the selected vertices in S_i. The only changed states are those of the vertices in X_i, and hence S_i is a (valid) partial solution. For a given vertex $v \in X_i$, the computation of its state *counts* (using the corresponding automaton) the global number of selected neighbors it has in $S_j \cup S_k = S_i$, by starting from its state in X_j and adding the number of selected neighbors it has in $S_k \setminus S_j$. $\square$

Theorem 2. *The problem* $\exists[\sigma, \rho]$-DOMINATING SET *can be solved in* $\mathcal{O}^*(s^{tw})$ *time on graphs of bounded tree-width, where* s *is a small polynomial in the total number of states of the two minimal automata enumerating* σ *and* ρ.

Proof (sketch). To obtain this time complexity, we use the same ideas as in [7], using transformations between states sets and taking advantage of fast subset convolution [1].

Suppose $\mathcal{Q}_\sigma = \{\sigma_0, \ldots, \sigma_a, \ldots, \sigma_p\}$ and $\mathcal{Q}_\rho = \{\rho_0, \ldots, \rho_b, \ldots, \rho_q\}$, where $\{\sigma_0, \ldots, \sigma_a\}$ is the aperiodic subset of σ, and $\{\rho_0, \ldots, \rho_b\}$ the aperiodic subset of ρ. Before the operation on a *join* node i with $|X_i| = n_i$, we transform the set of states $\{\sigma_0, \ldots, \sigma_a, \ldots, \sigma_p\}$ to $\{\sigma_0, \sigma_{\leq 1}, \ldots, \sigma_{\leq a}, \sigma_{a+1}, \ldots, \sigma_p\}$, and $\{\rho_0, \ldots, \rho_b, \ldots, \rho_q\}$ to $\{\rho_0, \rho_{\leq 1}, \ldots, \rho_{\leq b}, \rho_{b+1}, \ldots, \rho_q\}$. This transformation and its converse can be done in $\mathcal{O}^*(((p-a)^2 + (q-b)^2 + a + b)^{n_i})$ time each. Note that for finite or cofinite sets σ and ρ, we obtain an $\mathcal{O}^*((p+q)^{n_i})$ time complexity.

Recall that $n_i \leq tw$, and there are at most $\mathcal{O}(n)$ nodes in the nice tree-decomposition. As the time complexity of the operation on *join* nodes is the most costly, the overall time complexity of the algorithm follows. $\square$

C Parameterized complexity of $[\sigma, \rho]$-Dominating Set on general cases of σ and ρ

Throughout this paper, we study the parameterized complexity of $\exists[\sigma, \rho]$-Dominating Set for some special cases of σ and ρ. In this section, we give two general results on the parameterized complexity of $\exists[\sigma, \rho]$-Dominating Set and k-$[\sigma, \rho]$-Dominating Set for any recursive sets σ and ρ.

Theorem 3. *Let σ and ρ be two recursive sets of integers for which the membership of any integer t can be computed in polynomial time. Then $\exists[\sigma, \rho]$-*Dominating Set *(as well as minimization and maximization) is in* XP *when parameterized by the tree-width of the input graph.*

Proof. Let $G = (V, E)$ be the input graph. Indeed, any $[\sigma, \rho]$-dominating set will be of cardinality at most $|V|$. Hence we can consider the problem $[\sigma', \rho']$-Dominating Set where $\sigma' = \sigma \cap \{0, \ldots, |V|\}$ and $\rho' = \rho \cap \{0, \ldots, |V|\}$ are both finite. Note that σ' and ρ' can be computed in polynomial time, as the membership in σ or ρ of any integer t is required to be computable in $\mathcal{O}(\text{poly}(t))$ time. Using the algorithm given in Appendix B, $\exists[\sigma, \rho]$-Dominating Set can be solved in XP time when parameterized by the tree-width of the input graph.

Theorem 4. *Let σ and ρ be two recursive sets of integers. Then k-$[\sigma, \rho]$-*Dominating Set *is* FPT *when parameterized both by the tree-width of the input graph and the maximum size k of a solution.*

Proof. In k-$[\sigma, \rho]$-Dominating Set, we ask for a $[\sigma, \rho]$-dominating set of cardinality at most k. Hence every vertex of the input graph will have at most k neighbors in the $[\sigma, \rho]$-dominating set, so we can reduce this problem to $[\sigma', \rho']$-Dominating Set where $\sigma' = \sigma \cap \{0, \ldots, k\}$ and $\rho' = \rho \cap \{0, \ldots, k\}$ are both finite. If $a(k)$ (resp. $b(k)$) denotes the maximum time needed to decide whether $t \in \sigma$ (resp. $t \in \rho$) for $t \leq k$, then σ' and ρ' can be computed in $\mathcal{O}(k \cdot a(k))$ time (not depending on the size of the input graph) and hence in polynomial time. Using the algorithm given in Appendix B, k-$[\sigma, \rho]$-Dominating Set can be solved in FPT time when parameterized both by the tree-width of the input graph and the maximum size k of a solution.

Appendix References

1. Björklund, A., Husfledt, T., Kaski, P., Koivisto, M.: Fourier meets Möbius: fast subset convolution. In: STOC'07, Proceedings of the 39th Annual ACM Symposium on the Theory of Computing, San Diego, California (June 2007) 67–74
2. Bodlaender, H.L.: A linear-time algorithm for finding tree-decompositions of small treewidth. SIAM Journal on Computing **25** (1996) 1305–1317
3. Courcelle, B.: The expression of graph properties and graph transformations in monadic second-order logic. In: Handbook of Graph Grammars and Computing by Graph Transformation. Volume 1., River Edge, NJ, World Scientific (1997) 313–400
4. Courcelle, B., Makowsky, J.A., Rotics, U.: On the fixed parameter complexity of graph enumeration problems definable in monadic second-order logic. Discrete Applied Mathematics **108**(1–2) (2001) 23–52
5. Kloks, T.: Treewidth. Computations and approximations. Volume 842 of Lecture Notes in Computer Science. Springer (1994)
6. Matos, A.B.: Periodic sets of integers. Theoretical Computer Science **127** (1994) 287–312
7. van Rooij, J.M., Bodlaender, H.L., Rossmanith, P.: Dynamic programming on tree decompositions using generalised fast subset convolution. In: ESA'09, Proceedings of the 17th Annual European Symposium on Algorithms. Volume 5757 of Lecture Notes in Computer Science., Copenhagen, Denmark, Springer (September 2009) 566–577